ARMY RESEARCH LABORATORY

Parallel Numerical Computations of Projectile Flow Fields

by Jubaraj Sahu, Karen R. Heavey,
Daniel Pressel, and Surya Dinavahi

ARL-TR-2019 July 1999

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5066

ARL-TR-2019 **July 1999**

Parallel Numerical Computations of Projectile Flow Fields

Jubaraj Sahu and Karen R. Heavey
Weapons and Materials Research Directorate, ARL

Daniel Pressel
Corporate Information and Computing Directorate, ARL

Surya Dinavahi
Mississippi State University

Abstract

A time-marching, Navier-Stokes code, successfully used over a decade for projectile aerodynamics, was chosen as a test case and optimized to run on modern reduced instruction set computer (RISC)-based parallel computers. The parallelized version of the code has been used to compute the axisymmetric and three-dimensional (3-D) turbulent flow over a number of projectile configurations at transonic and supersonic speeds. In most of these cases, these results were then compared to those obtained with the original version of the code on a Cray C-90. Both versions of the code produced the same qualitative and quantitative results. Considerable performance gain was achieved by the optimization of the serial code on a single processor. Parallelization of the optimized serial code, which uses loop-level parallelism, led to additional gains in performance. The original algorithm remained unchanged. Recent runs on a 128-processor Origin 2000 have produced speedups in the range of 10–26 over that achieved when using a single processor on a Cray C-90. The original algorithm remained unchanged. Computed surface pressures were compared with the experimental data and were generally found to be in good agreement with the data.

Table of Contents

INTENTIONALLY LEFT BLANK.

List of Figures

INTENTIONALLY LEFT BLANK.

1. Introduction

Advancements in computer technology and state-of-the-art numerical procedures enable one to find solutions to complex time-dependent problems associated with projectile aerodynamics, store separation from fighter planes, and other multibody systems. Application of computational fluid dynamics (CFD) to multibody configurations has proven to be a valuable tool in evaluating potential new designs. Although the computational results obtained are encouraging and valuable, the computer central processing unit (CPU) time required for each time-dependent calculation is immense, even for axisymmetric flows, with three-dimensional (3-D) calculations being worse. This problem becomes even more extreme when one looks at the turnaround time. These times must be reduced at least an order of magnitude before this technology can be used routinely for the design of multibody projectile systems. This is also true for numerical simulation of single projectile-missile configurations, which are, at times, quite complex and require large computing resources. The primary technical challenge is to effectively utilize new advances in computer technology in order to significantly reduce run time and to achieve the desired improvements in the turnaround time.

The U.S. Department of Defense (DOD) is actively upgrading its high-performance computing (HPC) resources through the DOD High-Performance Computing Modernization Program (HPCMP). The goal of this program is to provide the research scientists and engineers with the best computational resources (networking, mass storage, and scientific visualization) for improved design of weapon systems. The program is designed to procure state-of-the-art computer systems and support environments. One of the initiatives of the DOD HPCMP is the Common High-Performance Computing Software Support Initiative aimed at developing application software for use with systems being installed. This program covers 10 computational technology areas (CTAs) that have been deemed crucial in the DOD science and engineering community. One of the CTAs is CFD. A major portion of this effort has to do with developing software to run on the new scalable systems, since much of the existing code was developed for vector systems. One of the codes that was selected for this effort is the F3D [1, 2] code, which was originally developed at NASA Ames Research Center with subsequent modifications made

at the U.S. Army Research Laboratory (ARL). This code is a Navier-Stokes solver capable of performing implicit and explicit calculations. It has been extensively validated and calibrated for many applications in the area of projectile aerodynamics for over a decade. As such, there was a strong interest in porting this code to the new environments. A key reason for choosing this flow solver is its proven ability to compute the flow field for projectile configurations using Navier-Stokes computational techniques [3–6]. The same flow solver has also been used to compute 3-D flow over various spinning and nonspinning projectile configurations. Computed result (including axial force, normal force, pitching moment, and Magnus force and moment) obtained with this code compared favorably with experimental and flight test data.

The key breakthrough was the realization that many of the new systems seemed to lend themselves to the use of loop-level parallelism. This strategy offered the promise of allowing the code to be parallelized with absolutely no changes to the algorithm. This paper describes the solution technique, parallelization of the code, and its application to Army projectile configurations.

2. Solution Technique

2.1 Governing Equations. The complete set of 3-D, time-dependent, generalized geometry, Reynolds-averaged, thin-layer Navier-Stokes equations is solved numerically to obtain a solution to this problem and can be written in general spatial coordinates ξ, η, and ζ as follows [1]:

$$\partial_\tau \hat{q} + \partial_\xi \hat{F} + \partial_\eta \hat{G} + \partial_\zeta \hat{H} = Re^{-1} \partial_\zeta \hat{S} \ , \tag{1}$$

In equation (1), $\hat{q}$ contains the dependent variables: density, three velocity components, and energy. The thin-layer approximation is used here, and the viscous terms involving velocity gradients in both the longitudinal and circumferential directions are neglected. The viscous terms are retained in the normal direction, ζ and are collected into the vector $\hat{S}$. Similar thin-layer approximation is also used in the other directions when needed.

2.2 Numerical Technique. The implicit, approximately factored scheme for the thin-layer Navier-Stokes equations using central differencing in the η and ζ directions and upwinding in ξ is written in the following form [1]:

$$\left[I + i_b h\delta_\xi^b\left(\hat{A}^+\right)^n + i_b h\delta_\zeta \hat{C}^n - i_b h Re^{-1}\bar{\delta}_\zeta J^{-1}\hat{M}^n J - i_b D_i\big|_\zeta\right]$$

$$\times\left[I + i_b h\delta_\xi^f\left(\hat{A}^-\right)^n + i_b h\delta_\eta \hat{B}^n - i_b D_i\big|_\eta\right]\Delta\hat{Q}^n$$

$$= -i_b\Delta t\left\{\delta_\xi^b\left[\left(\hat{F}^+\right)^n - \hat{F}_\infty^+\right] + \delta_\xi^f\left[\left(\hat{F}^-\right)^n - \hat{F}_\infty^-\right] + \delta_\eta\left(\hat{G}^n - \hat{G}_\infty\right)\right.$$

$$\left. + \delta_\zeta\left(\hat{H}^n - \hat{H}_\infty\right) - Re^{-1}\bar{\delta}_\zeta\left(\hat{S}^n - \hat{S}_\infty\right)\right\} - i_b D_e\left(\hat{Q}^n - \hat{Q}_\infty\right), \qquad (2)$$

where $h = \Delta t$ or $(\Delta t)/2$and the free-stream base solution is used. Here, δ is typically a three-point second-order-accurate central difference operator, $\bar{\delta}$ is a midpoint operator used with the viscous terms, and the operators δ_ξ^b and δ_ξ^f are backward and forward three-point difference operators. The flux $\hat{F}$ has been eigensplit, and the matrices $\hat{A}$, $\hat{B}$, $\hat{C}$, and $\hat{M}$ result from local linearization of the fluxes about the previous time level. Here, J denotes the Jacobian of the coordinate transformation. Dissipation operators D_e and D_i are used in the central space differencing directions.

2.3 Chimera Composite Grid Technique. The Chimera [7–9] overset grid scheme is a domain decomposition approach where a full configuration is meshed using a collection of independent overset grids. This allows each component of the configuration to be gridded separately and overset into a main grid. Overset grids are not required to join in any special way. Usually, a major grid covers the entire domain or a grid is generated about a dominant body section. Minor grids are generated about the rest of the bodies or sections. Because each component grid is generated independently, portions of one grid may be found to lie within a solid boundary contained within another grid. Such points lie outside the computational domain and are excluded from the solution process. Equation (2) has been modified for Chimera overset grids by the introduction of the flag i_b to achieve just that. This i_b array accommodates the

possibility of having arbitrary holes in the grid. The i_b array is defined such that $i_b = 1$ at normal grid points and $i_b = 0$ at hole points. Thus, when $i_b = 1$, equation (2) becomes the standard scheme. But, when $i_b = 0$, the algorithm reduces to $\Delta\hat{Q}^n = 0$ or $\hat{Q}^{n+1} = \hat{Q}^n$, leaving $\hat{Q}$ unchanged at hole points. The set of grid points that form the border between the hole points and the normal field points are called intergrid boundary points. These points are updated by interpolating the solution from the overset grid that created the hole. Values of the i_b array and the interpolation coefficients needed for this update are provided by a separate algorithm [7]. The Chimera procedure reduces a complex problem into a number of simpler subproblems. Computations are performed on each grid separately. A major part of the Chimera overset grid approach is the information transfer from one grid into another by means of the intergrid boundary points.

2.4 Boundary Conditions. For simplicity, most of the boundary conditions have been imposed explicitly. An adiabatic wall boundary condition is used on the body surface, and the no-slip boundary condition is used at the wall. The pressure at the wall is calculated by solving a combined momentum equation. Free-stream boundary conditions are used at the inflow boundary as well as at the outer boundary. A symmetry boundary condition is imposed at the circumferential edges of the grid, while a simple extrapolation is used at the downstream boundary. A combination of symmetry and extrapolation boundary condition is used at the center line (axis). For supersonic flows, a nonreflection boundary condition is used at the outer boundary. For overset grids, the outer boundary of the component grids completely lies within the background projectile grid and, thus, gets its flow-field information interpolated from the projectile grid.

3. Parallelization Methodology

Many modern parallel computers are now based on high-performance reduced instruction set computer (RISC) processors. There are two important conclusions that one can reach from this observation: (1) in theory, there are many cases in which it will no longer be necessary to use

over 100 processors in order to meet the user's needs and, (2) if the theory is to be met, one must achieve a reasonable percentage of the peak processing speed of the processors being used. Additionally, the first conclusion allows for the use of alternative architectures and parallelization techniques that might support only a limited degree of parallelism (e.g., 10–100 processors). Based on this reevaluation, some important conclusions were reached.

(1) In using traditional parallel algorithms and techniques, using significantly fewer processors can

   (a) decrease the system cost,

   (b) increase the reliability of the system,

   (c) decrease the extent to which the efficiency of the algorithm is degraded,

   (d) decrease the percentage of the run time spent passing messages, and

   (e) decrease the effect of Amdahl's Law.

(2) Possibly of even greater significance was the observation that, with loop-level parallelism, it is possible to avoid many of the problems associated with parallel programming altogether. This is not a new observation, but only now is it starting to be a useful one. The key things that changed are that

   (a) loop-level parallelism is frequently restricted to using modest numbers of processors and the processors therefore have to be fast enough to achieve an acceptable level of performance;

   (b) loop-level parallelism will, in general, try and use the same sources of parallelism used to produce a vectorizable code (this makes it difficult to efficiently use this type of parallelism on a machine equipped with vector processors); and

(c) it is difficult to make efficient use of loop-level parallelism on anything but a shared-memory architecture, and, only recently, have vendors started to ship shared-memory architectures based on RISC processors with aggregate peak speeds in excess of a few giga-floating-point operations per second (GFLOPS).

By combining aggressive serial optimizations with loop-level parallelization of vectorizable loops (some loop interchanging was also required), all of the design goals were met.

4. Results

4.1 Supersonic Flow Over a Missile Body. A generic missile configuration was used for many of the tests on the parallelized code. In these tests, a one-million-point grid (see Figure 1) was used to check the accuracy of the results. The computed results obtained with the parallelized code were compared with those obtained using the vectorized code on a Cray C-90. These computed results were compared with the experimental data obtained at the Defense Research Agency (DRA) [10], UK, for the same configuration and test conditions. Typically, computation on the C-90 used 18 MW (148 MB) of memory and 7.5 hr of CPU time. Once the accuracy of the computed result was verified, performance studies were carried out for grid sizes ranging from 1 to 53 million grid points. Figure 2 shows the computed pressure contours for Mach number, $M = 2.5$ and angle of attack, $\alpha = 14°$ for the 1-million-grid-point case. It shows the computed pressure contours for both windside (bottom) and leeside (top). Computed pressures were obtained at 1,800 time steps using both the Power Challenge Array (PCA) and the C-90. Both solutions produce identical results and show the expected shock wave flow features. Figure 3 shows the circumferential surface pressure distributions of the missile at a selected longitudinal station. Computed results from both versions of the code are shown to lie on top of one another and, thus, are in excellent agreement. Both computed and experimental results show the same trends (i.e., higher surface pressure on the windside and low pressure on the leeside).

These results were obtained using a highly efficient serial algorithm as the starting point, taking great care not to make any changes to the algorithm. Initial efforts to run the vector

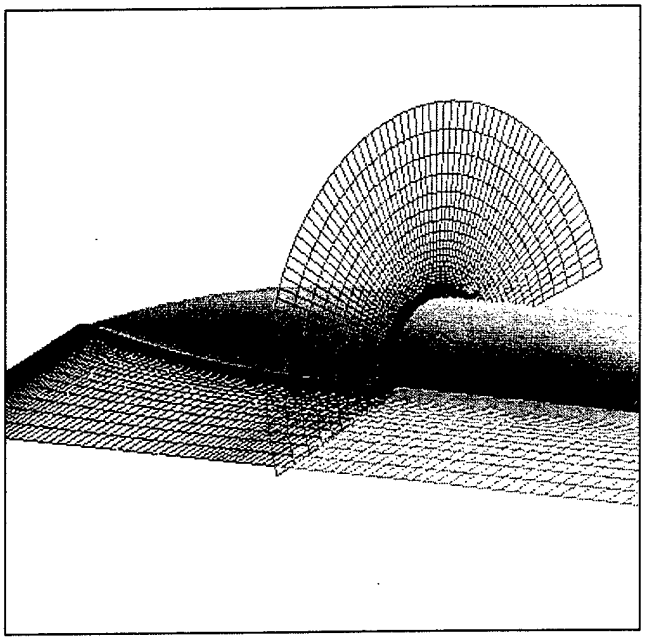

Figure 1. Computational Grid.

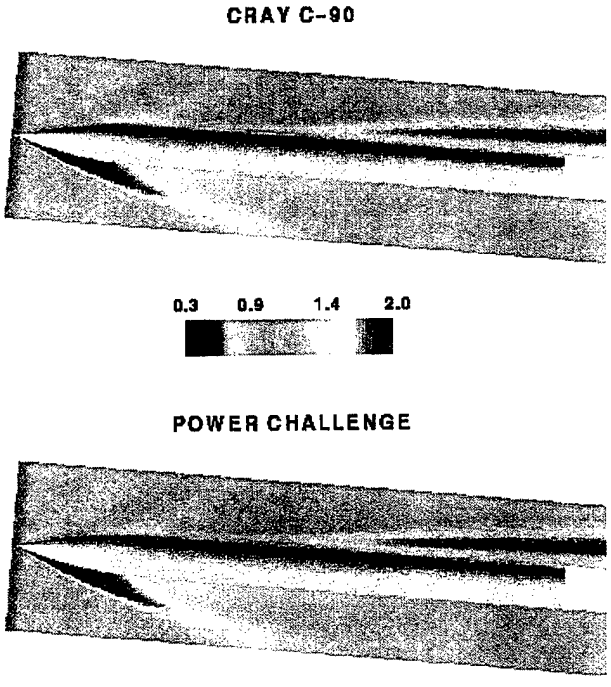

Figure 2. Comparison of Computed Pressure Contours on C-90 and PCA.

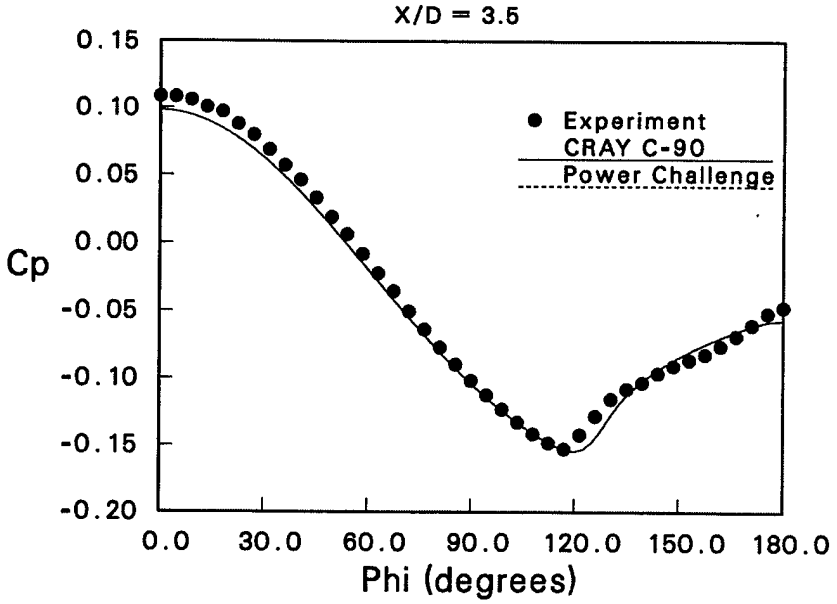

Figure 3. Surface Pressure Comparison.

optimized version of this code on one processor of a Silicon Graphics (SGI) Power Challenge (75-MHz R8000 processor) proved to be extremely disappointing. After aggressively tuning the code for a low-cache miss rate and good pipeline efficiency, a factor of 10 improvement in the serial performance of this code was achieved. At this point, the percentage of peak performance from the RISC-tuned code using one processor on the SGI Power Challenge was the same as the vector-tuned code on one processor of a Cray C-90. A key enabling factor was the observation that processors with a large external cache (e.g., 1–4 MB in size) could enable the use of optimization strategies that simply were not possible on machines like the Cray T3D and Intel Paragon, which only have 16 KB of cache per processor. This relates to the ability to size scratch arrays so that they will fit entirely in the large external cache. This can reduce the rate of cache misses associated with these arrays, which miss all the way back to main memory, to less than 0.1% (the comparable cache miss rates for machines like the Cray T3D and Intel Paragon could easily be as high as 25%).

While the effort to tune the code was nontrivial, the initial effort to parallelize the code was already showing good speedup on 12 processors (the maximum number of processors available in one Power Challenge at that time). Additional efforts extended this work to larger numbers of processors on a variety of systems. Most recently, work has been performed on 64- and

128-processor SGI Origin 2000s (the latter is an experimental system located at the Naval Research Laboratory in Washington, DC). This work has extended the range of problem sizes up to 53 million grid points spread between just three zones and up to 115 processors on the larger machine (due to the stair-stepping effect, the problem sizes run on this machine were not expected to get any additional benefit from using 116B128 processors).

Figure 4 shows the performance results for a data set. All results have been adjusted to remove start-up and termination costs. The latest results show a factor of 900B1,000 speedup from the original runs made using one processor of the Power Challenge with the vector-optimized code (the corresponding increase in processing power was less than a factor of 160). Additionally, speedups as high as 26.1 relative to one processor of a C-90 have been achieved. Since the numerical algorithm was unchanged, this represents a factor of 26.1 increase in the speed at which floating-point operations were performed and, consequently, the wall clock time required for a converged solution decreased by the same factor. In a production environment, such as is found at the four Major Shared Resource Centers (MSRCs) set up by the DOD HPCMP, these results represent an opportunity to significantly improve the job throughput. These results clearly demonstrate that, when using the kinds of techniques described herein, it is possible to achieve high levels of performance with good scalability on at least some RISC-based, shared-memory, symmetric multiprocessors. It is also interesting to note that these results were obtained without the use of any assembly code or system-specific libraries and with relatively little help from the vendors.

4.2 Transonic Flow Over a Secant-Ogive Cylinder-Boattail (SOCBT) Projectile. The projectile modeled in this example consists of a three-caliber secant-ogive nose, a two-caliber cylinder, and a one-caliber 7° boattail. For this case, the base is not included and the boattail is extended as a sting. A schematic diagram of the projectile is shown in Figure 5. Computed surface pressure is compared to experimental surface pressure measurements made by Kayser and Whiton [11]. The computational grid used for this calculation was obtained using a hyperbolic grid generator. The grid consists of 128 longitudinal points and 56 radial points. There are three planes in the circumferential direction. The computational domain extends to

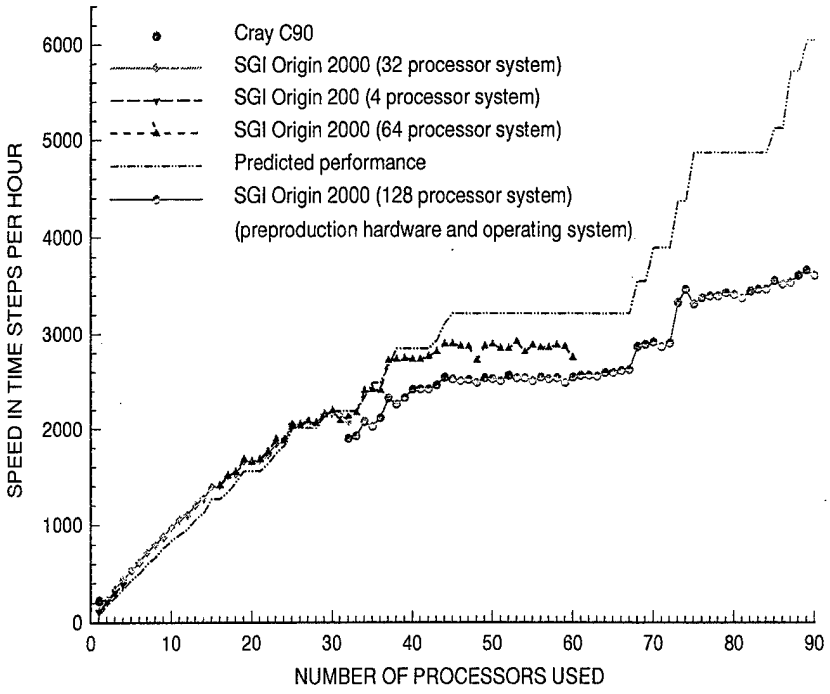

Figure 4. Performance Results for the One-Million-Grid-Point Data Set.

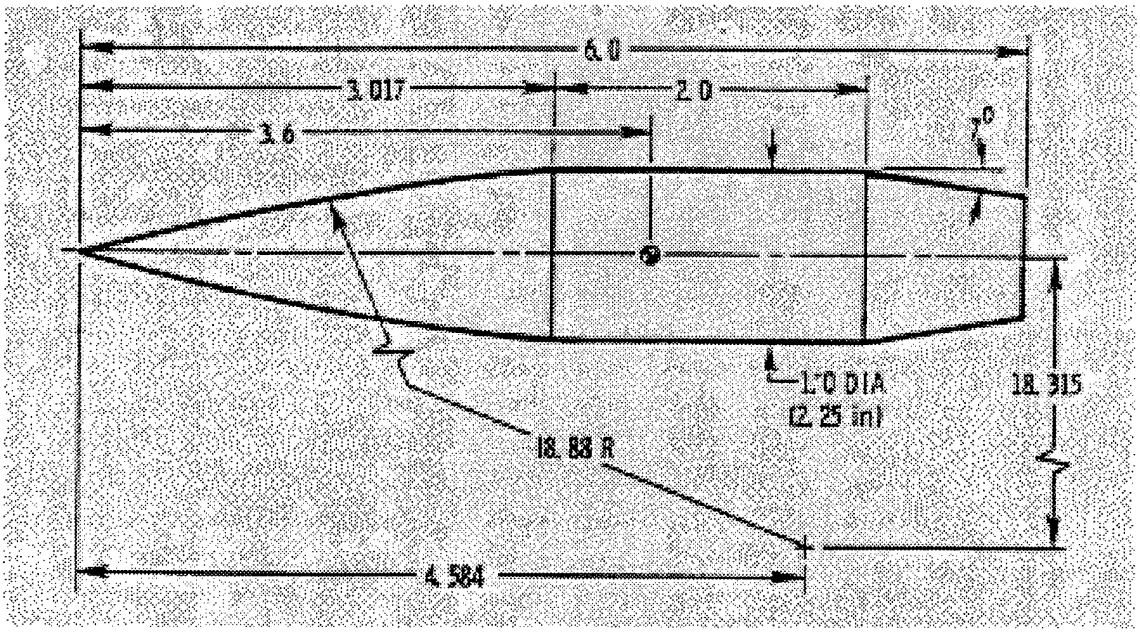

Figure 5. Schematic Diagram of SOCBT Projectile.

about 3.5 body lengths in front of the grid, in the radial direction, and behind the projectile. An expanded view of the grid near the projectile surface is shown in Figure 6. The grid points are clustered in the longitudinal direction at the ogive-cylinder and cylinder-boattail junctions. In

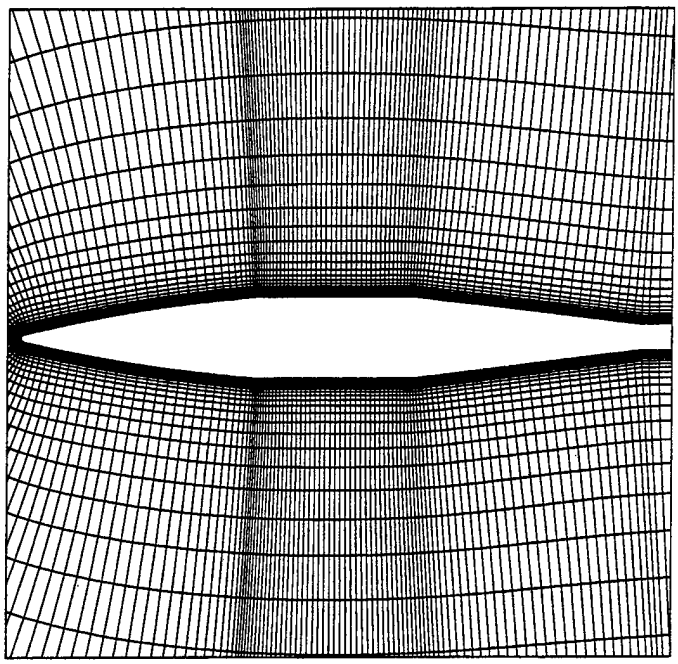

Figure 6. Computational Grid Near the Projectile.

the normal (radial) direction, the grid points are clustered near the body surface with a minimum spacing of 0.00002 and are stretched to the outer boundary. Figure 7 shows pressure contours for the converged solution. It shows the expansion of the flow at the ogive-cylinder and cylinder-boattail corners, as well as the location of the shock wave. Figure 8 shows a comparison of computed surface pressure with experimental data. The expansions at the projectile corners are clearly seen in the computation and are in good agreement with the experimental data.

4.3 Flow Over Segments. This multibody problem involves the separation of two submunitions at a low transonic speed. Figure 9 shows the components of the projectile configuration. The interest here is the aerodynamic interference effect of the two submunitions in flight. Each submunition is a right circular cylinder of length-to-diameter ratio of 1.38. The inflow, far-field, and outflow boundaries are placed far enough for computation of transonic flows. The complete grid consists of three zones, with approximately 83,000 grid points. Each grid section was obtained separately and then appended to provide the full grid. The Cartesian

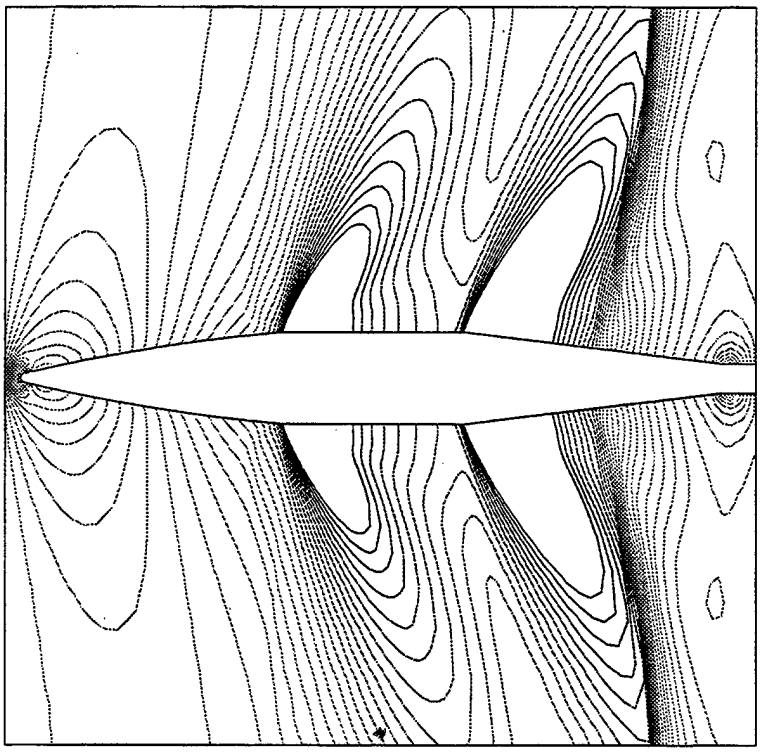

Figure 7. Computed Pressure Contours.

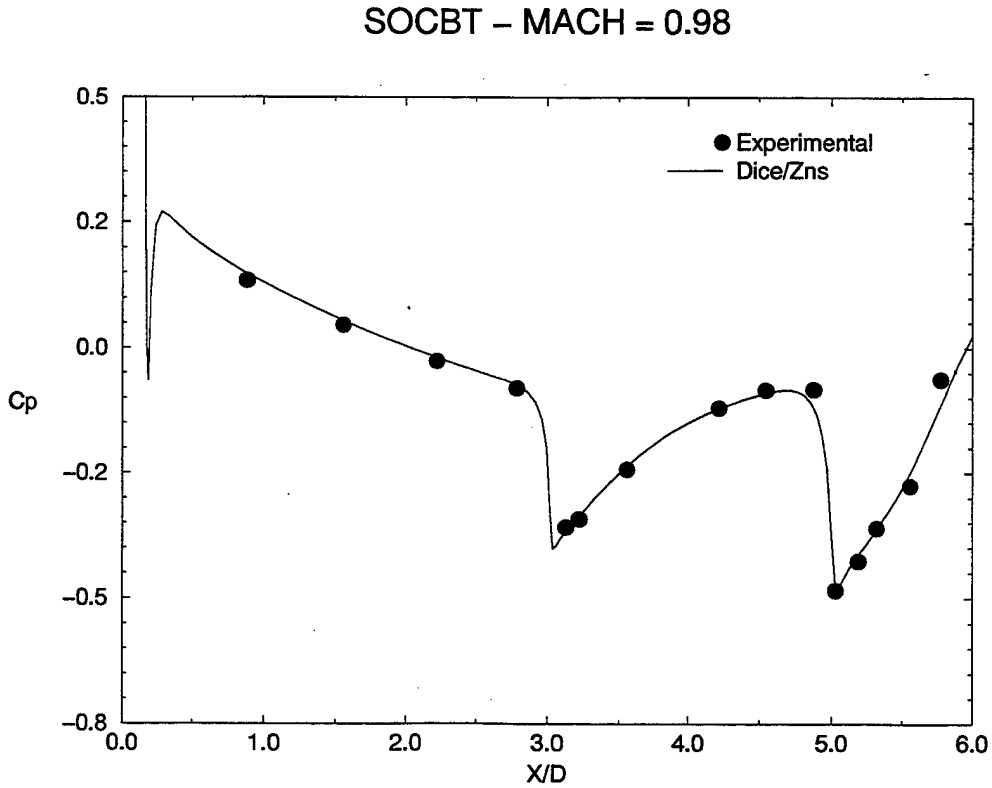

Figure 8. Comparison of Surface Pressure.

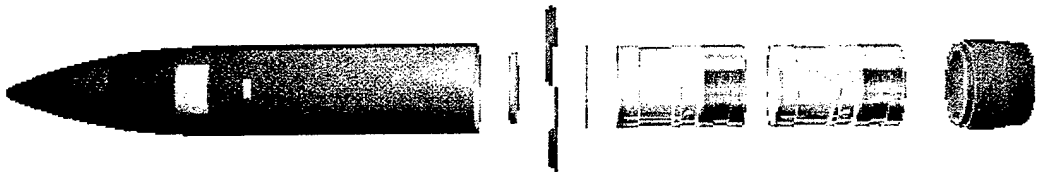

Figure 9. Projectile Configuration.

grid, which forms the background domain, was obtained algebraically and consists of 200 × 3 × 90 points in the axial, circumferential, and normal directions. A body-conforming grid (164 × 3 × 30) was obtained for each submunition using a hyperbolic grid generator. These submunition grids are then overset on the background Cartesian grid to form the composite mesh. An expanded view of the composite overset mesh system is shown in Figure 10 for the multibody separation problem. The first (leading) submunition grid is a minor grid, as is the second (trailing) submunition grid. The minor grids are completely overlapped by the major grid; thus, their outer boundaries can obtain information by interpolation from the major grid. Similar data transfer or communication is needed from the minor grids to the major grid. However, a natural outer boundary that overlaps the two submunition grids does not exist. The Chimera technique creates an artificial boundary (also known as a hole boundary) between grids that provides the required path for information transfer from the minor submunition grids to the background grid. The resulting hole region is excluded from the flow-field solution in the background grid. This case (see Figure 10) corresponds to a separation distance of one caliber between the two submunitions.

Figure 11 shows the components of another multibody projectile configuration, which included a design modification for the second submunition. A thin fin is added at the back of the second submunition to provide more drag during the separation process. The same Cartesian background is used in this case. The second submunition grid was obtained separately using a hyperbolic grid generator. It consists of 228 × 3 × 30 points. The major grid or the background grid is easily generated independently of the minor grid (the grid for the submunitions). The composite overset mesh system for this case is shown in Figure 12. For moving-body problems, both minor grids (shown in Figure 12) can move with the submunitions as they separate from

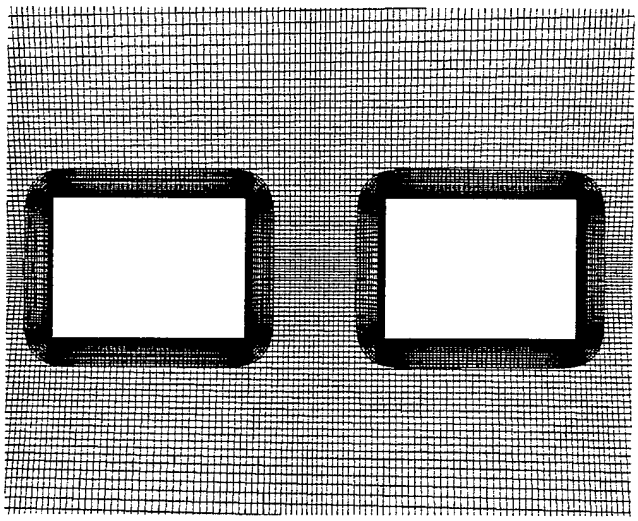

Figure 10. Grids for Two Submunitions.

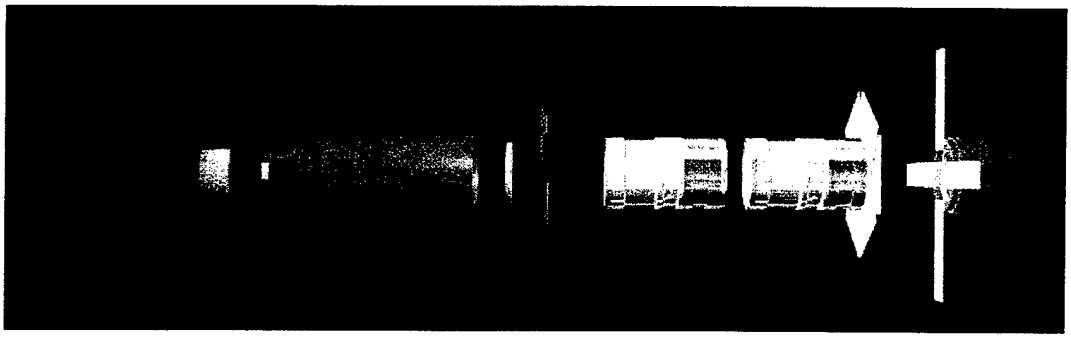

Figure 11. Modified Projectile Configuration.

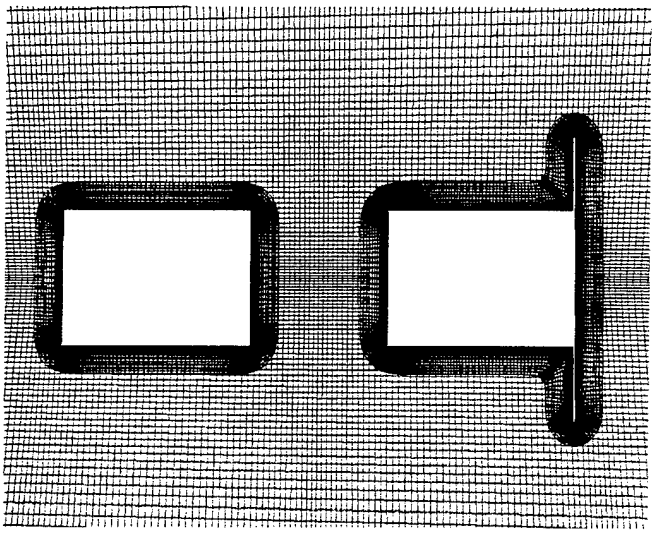

Figure 12. Grids for Modified Configuration.

each other. Again, there is no need to generate new grids for the submunitions during the dynamic process. An advantage of the Chimera technique is that it allows computational grids to be obtained for each body component separately and, thus, makes the grid generation process easier. Grid points are clustered near the submunition surfaces to capture the viscous boundary layers.

Numerical computations were performed for these configurations at Mach number, $M_\infty = 0.80$, and angle of attack, $\alpha = 0°$. Results are presented for both the original and the modified designs. Figure 13 shows the Mach number contours for the submunitions for the original configuration. The flow field is unsteady, and the second submunition is completely submerged in the wake of the first submunition. The pressure behind the first submunition is lower than the pressure ahead of it and, therefore, as expected, results in positive drag. The pressure behind the second submunition is, however, higher than the pressure ahead of it and, therefore, results in negative drag. Since the drag for the first submunition is positive, it tends to slow its motion. The drag for the second submunition is negative, which results in it being pulled back toward the first submunition. This can lead to undesirable submunition collisions. To avoid the submunition collision, fins were added to the second submunition to provide added drag. The same Chimera composite overset grid approach was used to numerically model this modified configuration. Figure 14 shows the Mach number contours for the submunitions for the modified multibody design. As seen in this figure, the second submunition is, again, completely submerged in the wake of the first submunition. It also indicates that the fin affects the flow field for the first submunition. The drag for the second submunition for the modified design case is larger than that obtained with the original design. As separation distance is increased between the submunitions, the drag for the second submunition should similarly go up. This increase in drag for the finned configuration allows the submunitions to continually separate and not come back and collide.

4.4 Flow Over a Projectile-Sabot System. Another multibody problem involves the separation of sabots from a projectile (see Figure 15). The aerodynamic interference of the projectile and the sabot flow field is quite complex and involves 3-D shock-boundary layer

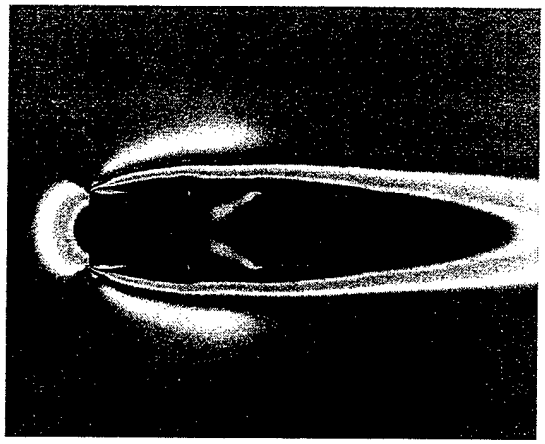

Figure 13. Mach Contours (Original Design).

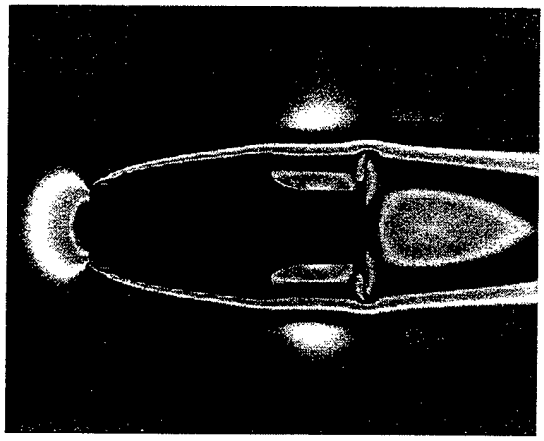

Figure 14. Mach Contours (Modified Design).

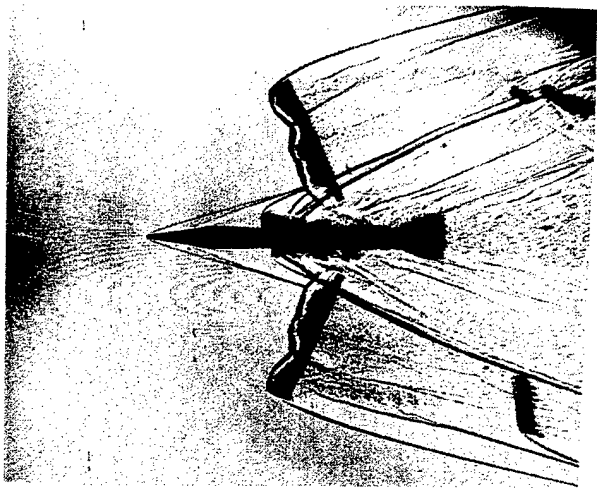

Figure 15. Shadowgraph of a Projectile-Sabot System.

interactions and separated flow regions. Again, with the overset grid approach, it is possible to use different grid topologies for the projectile and the sabot components, respectively. Figure 16 shows three overset meshes for a sabot angle of attack of 5°, 10°, and 15°. The same sabot grid used for the 5° angle-of-attack case is used for other sabot angles of attack without the need to regenerate new sabot grids. Figure 16 shows computational grids for the complete model, including the projectile and sabot. The projectile grid consists of three zones, with large number of grid points clustered near the sabot region. The first zone is a C-grid, while zones two and three are rectangular grids, for a total of approximately 800,000 grid points. The grids around the sabot also consist of three zones and were obtained using O-topology and rectangular topology. The main sabot grid consists of $94 \times 109 \times 39$ points in the axial, circumferential, and normal directions. The sabot component also required two other grids (a front grid and an aft grid). Both these grids are rectangular grids. The sabot grids were individually generated and then overset to form the complete grid system. In addition, there is a cover grid over the entire system. The computational grids shown here correspond to the pitch plane. The projectile grid serves as the main background grid for the computation. Steady-state numerical calculations have been performed for the projectile-sabot system at $M_\infty = 4.0$ and $\alpha = 0°$. Computational modeling is restricted to the symmetric sabot discard to save computer resources and time. The projectile is at zero angle of attack, and three sabots are discarded symmetrically following the same radial trajectory away from the projectile.

Computational studies have been completed for sabot angles of attack of 5°, 10°, and 15°. The projectile is at zero angle of attack for these three cases. As stated earlier, the background grid for the projectile remains the same. The sabot grids are, again, the same but have been moved to the new positions and orientations. Figure 17 shows the pressure contours for the projectile and sabot in the symmetry plane, for all three sabot orientations. This figure shows the interactions of the projectile and the sabot flow fields occurring at different longitudinal locations along the projectile. The computed pressure contours show the sabot shock impinging on the projectile, reflecting from the projectile surface. The shock impingement results in a higher pressure region on the projectile surface just downstream of the impingement point. As expected, the flow behind the base region of the sabot is a low-pressure region. As the sabot

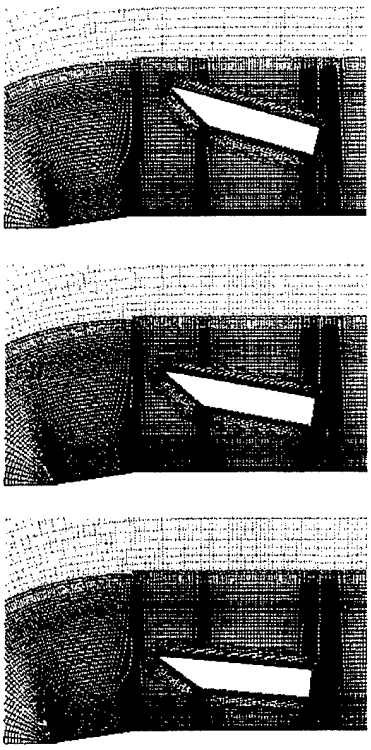

Figure 16. Grids for Three Complete Models.

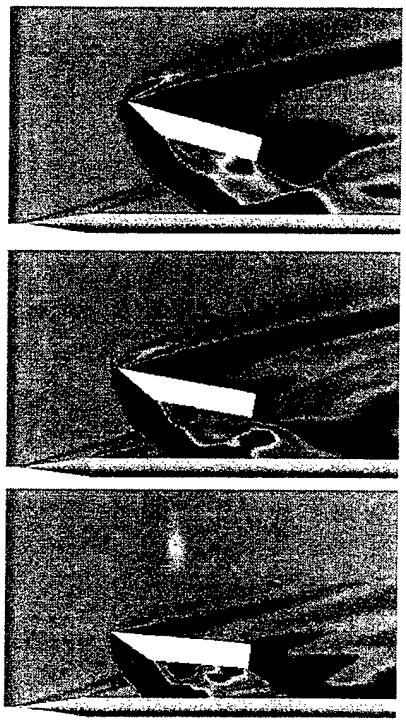

Figure 17. Computed Pressure Contours at 5°, 10°, and 15° Angle of Attack, Bottom and Top, M = 4.0.

angle of attack is increased, the sabot shock impingement point on the projectile is moved further downstream. For the 5° sabot angle of attack, the sabot shock impinges on the projectile, reflects from the projectile surface, and impinges back on the sabot. The reflected shock from the projectile surface is seen to just miss the base of the sabot for the 10° sabot angle-of-attack case and is even further away from the sabot base for the 15° sabot angle of attack case. The flow field in the base region of the sabot is also seen to change considerably with an increase in sabot angle of attack.

Figures 18 and 19 show computed surface pressures for the sabot and the projectile, respectively. These computed surface pressures correspond to the pitch plane and are compared with the experimental data [12]. The computed pressures on the bottom surface of the sabot are shown in Figure 18 and are generally found to be in agreement with the experimental data. Some discrepancies do exist in the comparison of sabot surface pressure for the 5° angle-of-attack case. Due to close proximity of the sabot to the projectile, the flow field is more complicated and includes complex shock-shock and shock-boundary layer interactions. Accurate computation of the resulting flow field is thus more difficult. Here, X/D = 0 corresponds to the nose of the projectile. Figure 19 shows the surface pressure distributions on the projectile in the pitch plane, for 5°, 10°, and 15° sabot angle-of-attack cases. Computed results are shown as a solid line and are compared with the experimental data shown in dark circles. As seen in this figure, the surface pressure is almost constant on the nose, which is followed by a pressure drop at the cylinder junction. This computed pressure drop at the cone-cylinder junction agrees well with the data at the 10° and 15° sabot angles of attack; however, the agreement is not as good for the 5° case. The predicted flow on the nose of the projectile corresponds to an undisturbed flow upstream of the shock impingement point. Clearly, the numerical results do not show the same extent of shock-boundary layer interactions observed experimentally. A large pressure increase due to the shock wave impinging on the projectile surface is seen in both computed and experimental data. The locations and magnitudes of the pressure peaks have been predicted fairly well.

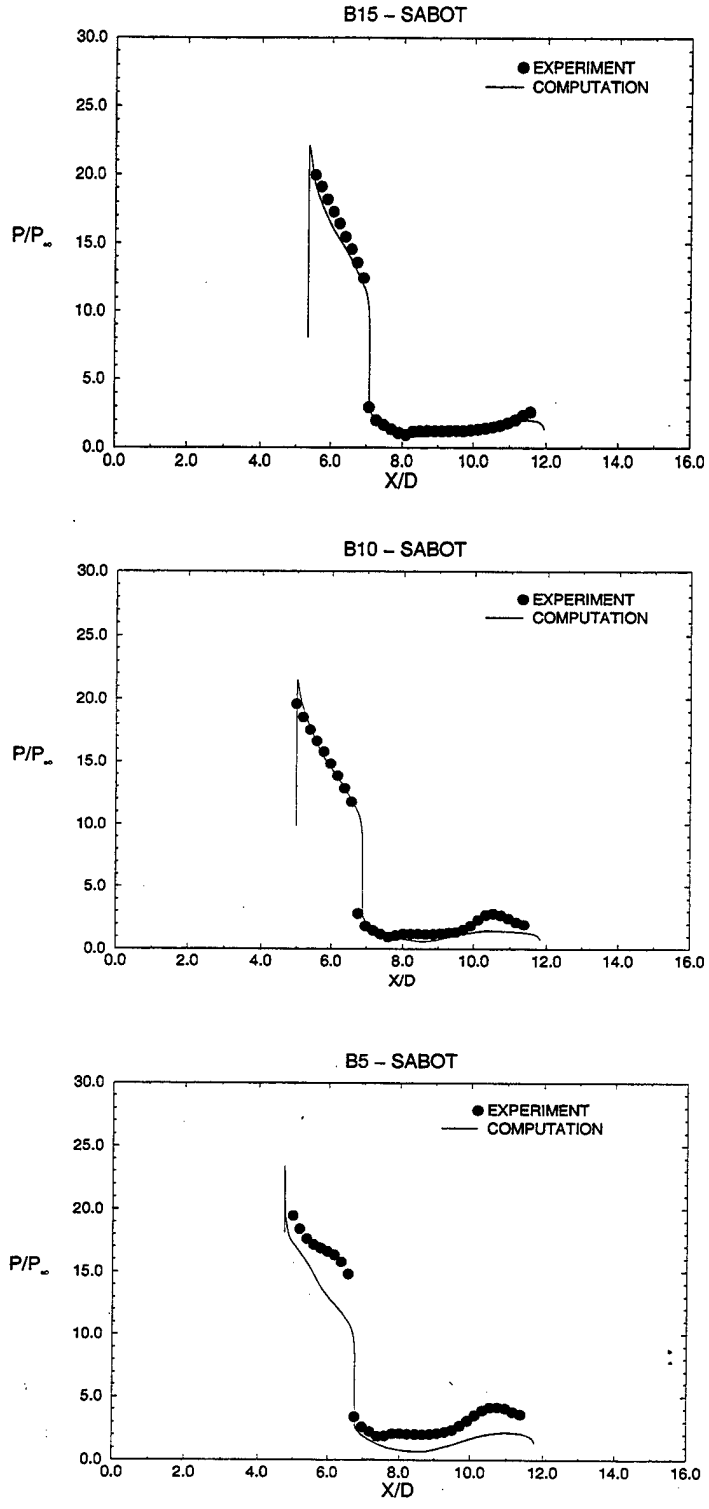

Figure 18. Sabot Surface Pressure Distributions.

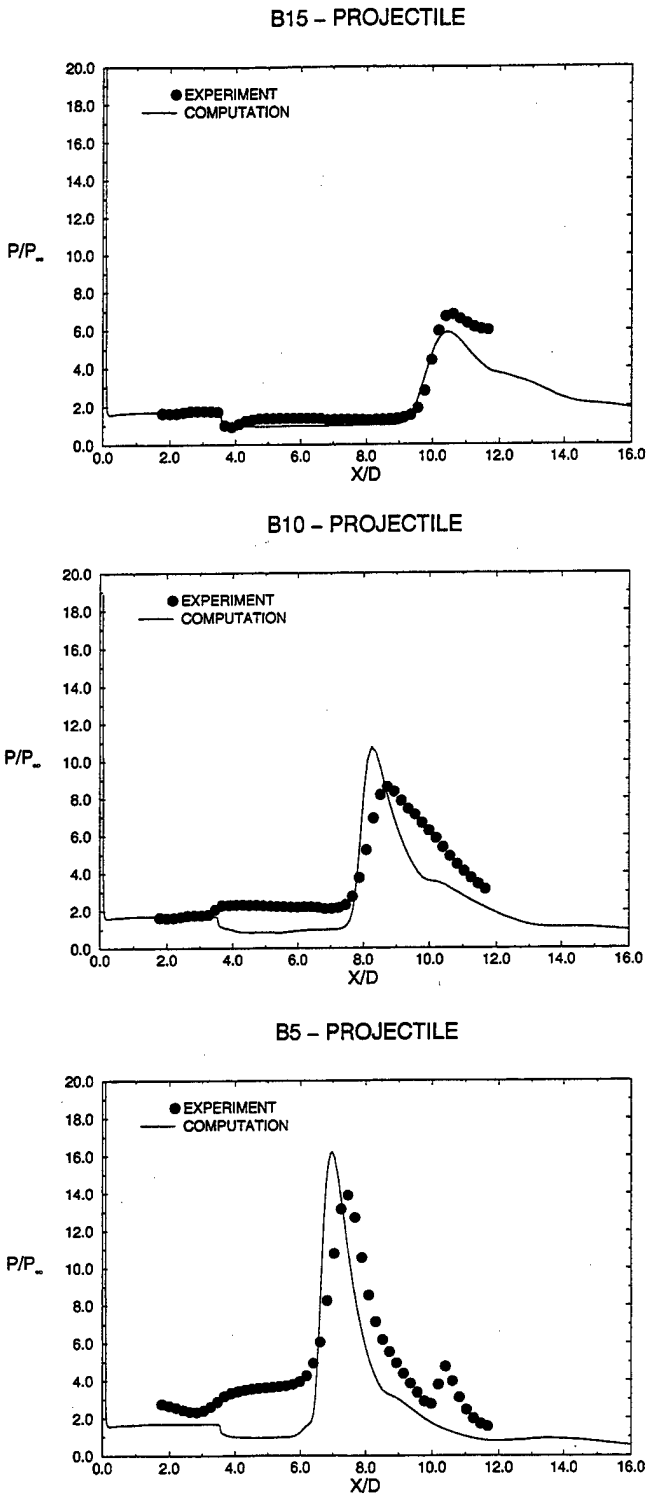

Figure 19. Projectile Surface Pressure Distributions.

5. Concluding Remarks

A time-marching, Navier-Stokes code, successfully used over a decade for projectile aerodynamics, was chosen as a test case and optimized to run on modern RISC-based parallel computers. The parallelized version of the code has been used to compute the axisymmetric and 3-D turbulent flow over a number of projectile configurations at transonic and supersonic speeds. In most of these cases, these results were then compared to those obtained with the original version of the code on a Cray C-90. Both versions of the code produced the same qualitative and quantitative results. Considerable performance gain was achieved by the optimization of the serial code on a single processor. Parallelization of the optimized serial code, which uses loop-level parallelism, led to additional gains in performance. The original algorithm remained unchanged. Recent runs on a 128-processor Origin 2000 have produced speedups in the range of 10–26 over that achieved when using a single processor on a Cray C-90. The original algorithm remained unchanged. Computed surface pressures were compared with the experimental data and were generally found to be in good agreement with the data.

6. References

1. Steger, J. L., S. X. Ying, and L. B. Schiff. "A Partially Flux-Split Algorithm for Numerical Simulation of Compressible Inviscid and Viscous Flows." *Proceedings of the Workshop on CFD*, Institute of Nonlinear Sciences, University of California, Davis, CA, 1986.

2. Sahu, J., and J. L. Steger. "Numerical Simulation of Transonic Flows." *International Journal for Numerical Methods in Fluids*, vol. 10, no. 8, pp. 855–873, 1990.

3. Sahu, J., C. J. Nietubicz, and J. L. Steger. "Navier-Stokes Computations of Projectile Base Flow With and Without Base Injection." *AIAA Journal*, vol. 23, no. 9, pp. 1348–1355, September 1985.

4. Sahu, J. "Numerical Computations of Supersonic Base Flow With Special Emphasis on Turbulence Modeling." *AIAA Journal*, vol. 32, no. 7, July 1994.

5. Sahu, J. "Numerical Computations of Transonic Critical Aerodynamic Behavior." *AIAA Journal*, vol. 28, no. 5, pp. 807–816, May 1990.

6. Sahu, J., and C. J. Nietubicz. "Application of Chimera Technique to Projectiles in Relative Motion." *AIAA Journal of Spacecraft and Rockets*, vol. 32, no. 5, September–October 1995.

7. Steger, J. L., F. C. Dougherty, and J. A. Benek. "A Chimera Grid Scheme." *Advances in Grid Generation*, ASME FED-5, K. N. Ghia and U. Ghia (editors), June 1983.

8. Benek, J. A., T. L. Donegan, and N. E. Suhs. "Extended Chimera Grid Embedding Scheme With Application to Viscous Flows." AIAA Paper No. 87-1126-CP, 1987.

9. Meakin, R. L. "Computations of the Unsteady Flow About a Generic Wing/Pylon/Finned-Store Configuration." AIAA 92-4568-CP, August 1992.

10. Birch, T. Private communication. Defense Research Agency, UK, 1997.

11. Kayser, L.D., and F. Whiton. "Surface Pressure Measurements on a Boattailed Projectile Shape at Transonic Speeds." ARBRL-MR-03161, US Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, 1982.

12. Lesage, F., and B. Girard. "Wind Tunnel and CFD Investigation of Aerodynamic Interactions During Sabot Separation." AIAA Paper No. 96-0193, January 1996.

INTENTIONALLY LEFT BLANK.

NO. OF COPIES ORGANIZATION

2 DEFENSE TECHNICAL INFORMATION CENTER
DTIC DDA
8725 JOHN J KINGMAN RD
STE 0944
FT BELVOIR VA 22060-6218

1 HQDA
DAMO FDQ
D SCHMIDT
400 ARMY PENTAGON
WASHINGTON DC 20310-0460

1 OSD
OUSD(A&T)/ODDDR&E(R)
R J TREW
THE PENTAGON
WASHINGTON DC 20301-7100

1 DPTY CG FOR RDA
US ARMY MATERIEL CMD
AMCRDA
5001 EISENHOWER AVE
ALEXANDRIA VA 22333-0001

1 INST FOR ADVNCD TCHNLGY
THE UNIV OF TEXAS AT AUSTIN
PO BOX 202797
AUSTIN TX 78720-2797

1 DARPA
B KASPAR
3701 N FAIRFAX DR
ARLINGTON VA 22203-1714

1 NAVAL SURFACE WARFARE CTR
CODE B07 J PENNELLA
17320 DAHLGREN RD
BLDG 1470 RM 1101
DAHLGREN VA 22448-5100

1 US MILITARY ACADEMY
MATH SCI CTR OF EXCELLENCE
DEPT OF MATHEMATICAL SCI
MADN MATH
THAYER HALL
WEST POINT NY 10996-1786

NO. OF COPIES ORGANIZATION

1 DIRECTOR
US ARMY RESEARCH LAB
AMSRL DD
J J ROCCHIO
2800 POWDER MILL RD
ADELPHI MD 20783-1197

1 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CS AS (RECORDS MGMT)
2800 POWDER MILL RD
ADELPHI MD 20783-1145

3 DIRECTOR
US ARMY RESEARCH LAB
AMSRL CI LL
2800 POWDER MILL RD
ADELPHI MD 20783-1145

ABERDEEN PROVING GROUND

4 DIR USARL
AMSRL CI LP (BLDG 305)

NO. OF COPIES	ORGANIZATION
2	USAF WRIGHT ARNTCL LABS AFWAL FIMG J SHANG N E SCAGGS WPAFB OH 45433-6553
1	CDR NSWC W YANTA CODE B40 DAHLGREN VA 22448-5100
1	CDR NSWC A WARDLAW CODE 420 INDIAN HEAD MD 20640-5035
4	DIR NATIONAL AERONAUTICS AND SPACE ADMINISTRATION LANGLEY RESEARCH CENTER TECH LIBRARY D M BUSHNELL M J HEMSCH J SOUTH LANGLEY STATION HAMPTON VA 23665
2	ARPA P KEMMEY J RICHARDSON 3701 NORTH FAIRFAX DR ARLINGTON VA 22203-1714
7	DIR NASA AMES RSRCH CNTR MS 227 8 L SCHIFF MS 258 1 T HOLST D CHAUSSEE M RAI P KUTLER P BUNING B MEAKIN MOFFETT FIELD CA 94035
1	USMA DEPT OF MECHANICS LTC ANDREW L DULL WEST POINT NY 10996

NO. OF COPIES	ORGANIZATION
7	CDR USA TACOM ARDEC AMSTA AR FSF T R DEKLEINE C NG R BOTTICELLI H HUDGINS J GRAU S KAHN W KOENIG BLDG 382 PICATINNY ARSENAL NJ 07806-5000
1	CDR USA TACOM AMSTA AR CCH B P VALENTI BLDG 65 S PICATINNY ARSENAL NJ 07806-5001
1	CDR USA ARDEC SFAE FAS SD M DEVINE PICATINNY ARSENAL NJ 07806-5001
1	CDR NSWC F MOORE DAHLGREN VA 22448
2	UNIV OF CALIFORNIA DAVIS DEPT OF MECHANICAL ENGNRNG H A DWYER M HAFEZ DAVIS CA 95616
1	AEROJET ELECTRONICS PLANT D W PILLASCH B170 DEPT 5311 PO BOX 296 1100 WEST HOLLYVALE STREET AZUSA CA 91702

NO. OF COPIES ORGANIZATION

3 AIR FORCE ARMAMENT LAB
AFATL FXA
S C KORN
B SIMPSON
D BELK
EGLIN AFB FL 32542-5434

1 MASSACHUSETTS INSTITUTE OF TECHNOLOGY
TECH LIBRARY
77 MASSACHUSETTS AVE
CAMBRIDGE MA 02139

1 GRUMANN AEROSPACE CORP
AEROPHYSICS RESEARCH DEPT
R E MELNIK
BETHPAGE NY 11714

2 MICRO CRAFT INC
J BENEK
N SUHS
207 BIG SPRINGS AVE
TULLAHOMA TN 37388-0370

1 LANL
B HOGAN
MS G770
LOS ALAMOS NM 87545

3 DIR SNL
DIV 1554
W OBERKAMPF
F BLOTTNER
DIV 1636
W WOLFE
ALBUQUERQUE NM 87185

1 NAVAL AIR WARFARE CENTER
D FINDLAY
MS 3 BLDG 2187
PATUXENT RIVER MD 20670

1 METACOMP TECHNOLOGIES INC
S R CHAKRAVARTHY
650 HAMPSHIRE ROAD
SUITE 200
WESTLAKE VILLAGE CA 91361-2510

NO. OF COPIES ORGANIZATION

2 ROCKWELL SCIENCE CENTER
S V RAMAKRISHNAN
V V SHANKAR
1049 CAMINO DOS RIOS
THOUSAND OAKS CA 91360

1 ADVANCED TECHNOLOGY CTR
ARVIN CALSPAN
AERODYNAMICS RESEARCH DEPT
M S HOLDEN
PO BOX 400
BUFFALO NY 14225

1 PENNSYLVANIA STATE UNIV
DEPT OF AEROSPACE ENGNRNG
G S DULIKRAVICH
UNIVERSITY PARK PA 16802

1 UNIV OF ILLINOIS AT URBANA CHAMPAIGN
DEPT OF MECHANICAL AND INDUSTRIAL ENGINEERING
J C DUTTON
URBANA IL 61801

1 UNIVERSITY OF MARYLAND
DEPT OF AEROSPACE ENGNRNG
J D ANDERSON JR
COLLEGE PARK MD 20742

1 UNIVERSITY OF NOTRE DAME
DEPT OF AERONAUTICAL AND MECHANICAL ENGINEERING
T J MUELLER
NOTRE DAME IN 46556

1 UNIVERSITY OF TEXAS
DEPT OF AEROSPACE ENGNRNG MECHANICS
D S DOLLING
AUSTIN TX 78712-1055

1 UNIVERSITY OF DELAWARE
DEPT OF MECHANICAL ENGNRNG
J MEAKIN
NEWARK DE 19716

NO. OF
COPIES ORGANIZATION

4 CDR USAAMCOM
AMSAM RD SS AT
E KREEGER
G LANDINGHAM
C D MIKKELSON
E VAUGHN
REDSTONE ARSENAL AL
35898-5252

1 CDR
USATACOM ARDEC
BLDG 162S
AMCPM DS MO
P J BURKE
PICATINNY ARSENAL NJ
07806-5000

3 CDR USA ARDEC
FIRING TABLES BRANCH
R LIESKE
R EITMILLER
F MIRABELLE
BLDG 120
ABERDEEN PROVING GROUND MD
21005-5066

ABERDEEN PROVING GROUND

31 DIR USARL
AMSRL WM
I MAY
L JOHNSON
AMSRL WM B
A W HORST JR
W CIPEIELLA
E M SCHMIDT
AMSRL WM BA
W D'AMICO
F BRANDON
T BROWN
AMSRL WM BE
G WREN
M NUSCA
J DESPIRITO
AMSRL WM BC
P PLOSTINS
D LYON
M BUNDY
G COOPER
J GARNER

NO. OF
COPIES ORGANIZATION

B GUIDOS
K HEAVEY
H EDGE
V OSKAY
A MIKHAIL
J SAHU
K SOENCKSEN
D WEBB
P WEINACHT
S WILKERSON
A ZIELINSKI
AMSRL WM BD
B FORCH
AMSRL WM BF
J LACETERA
AMSRL CI
C NIETUBICZ
AMSRL DI H
W STUREK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 1999	Final, February 1998–April 1999

4. TITLE AND SUBTITLE

Parallel Numerical Computations of Projectile Flow Fields

5. FUNDING NUMBERS

1L162628AH80

6. AUTHOR(S)

Jubaraj Sahu, Karen R. Heavey, Daniel Pressel, and Surya Dinavahi*

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
ATTN: AMSRL-WM-BC
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER

ARL-TR-2019

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

*Mississippi State University, MS 39762

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

A time-marching, Navier-Stokes code, successfully used over a decade for projectile aerodynamics, was chosen as a test case and optimized to run on modern reduced instruction set computer (RISC)-based parallel computers. The parallelized version of the code has been used to compute the axisymmetric and three-dimensional (3-D) turbulent flow over a number of projectile configurations at transonic and supersonic speeds. In most of these cases, these results were then compared to those obtained with the original version of the code on a Cray C-90. Both versions of the code produced the same qualitative and quantitative results. Considerable performance gain was achieved by the optimization of the serial code on a single processor. Parallelization of the optimized serial code, which uses loop-level parallelism, led to additional gains in performance. The original algorithm remained unchanged. Recent runs on a 128-processor Origin 2000 have produced speedups in the range of 10–26 over that achieved when using a single processor on a Cray C-90. The original algorithm remained unchanged. Computed surface pressures were compared with the experimental data and were generally found to be in good agreement with the data.

14. SUBJECT TERMS

computational fluid dynamics, projectile, parallel computations, transonic, supersonic flow

15. NUMBER OF PAGES

32

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

INTENTIONALLY LEFT BLANK.

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author ARL-TR-2019 (Sahu) Date of Report July 1999

2. Date Report Received _____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) _____

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) _____

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. _____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) _____

CURRENT ADDRESS

_____
Organization

_____
Name E-mail Name

_____
Street or P.O. Box No.

_____
City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD ADDRESS

_____
Organization

_____
Name

_____
Street or P.O. Box No.

_____
City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)

(DO NOT STAPLE)